INVENTOR-INFORMATION:

| NAME | CITY | STATE | COUNTRY | RULE-47 |
|---|---|---|---|---|
| Kunitake, Setsu | Nakai-machi | | JP | |
| Yamashita, Ichiro | Nakai-machi | | JP | |
| Kawabe, Shigehisa | Nakai-machi | | JP | |

US-CL-CURRENT: 707/517; 707/514, 707/531

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 9. Document ID: US 20010007987 A1

L9: Entry 9 of 11 File: PGPB Jul 12, 2001

PGPUB-DOCUMENT-NUMBER: 20010007987
PGPUB-FILING-TYPE: new-utility
DOCUMENT-IDENTIFIER: US 20010007987 A1

TITLE: Structured-document search apparatus and method, recording medium storing structured-document searching program, and method of creating indexes for searching structured documents

PUBLICATION-DATE: July 12, 2001

INVENTOR-INFORMATION:

| NAME | CITY | STATE | COUNTRY | RULE-47 |
|---|---|---|---|---|
| Igata, Nobuyuki | Kawasaki | | JP | |

US-CL-CURRENT: 707/3; 707/1

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 10. Document ID: US 6424979 B1

L9: Entry 10 of 11 File: USPT Jul 23, 2002

US-PAT-NO: 6424979
DOCUMENT-IDENTIFIER: US 6424979 B1

TITLE: System for presenting and managing enterprise architectures

DATE-ISSUED: July 23, 2002

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Livingston; Eric Bruce | Fairfax | VA | | |
| Vogt; Lisa Conners | Bethesda | MD | | |
| Gantz; Stephen David | Arlington | VA | | |
| Zipkin; David Samuel | Arlington | VA | | |
| Richards; Matthew Todd | Vienna | VA | | |
| Rocchio; Sheila Caroline | Arlington | VA | | |

ASSIGNEE-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY | TYPE CODE |
|---|---|---|---|---|---|
| American Management Systems, Inc. | Fairfax | VA | | | 02 |

APPL-NO: 09/ 223018
DATE FILED: December 30, 1998

PARENT-CASE:
CROSS REFERENCE TO RELATED APPLICATION This application is related to U.S. application entitled CONTENT MANAGEMENT SYSTEM having Ser. No. 09/222,831 by Baxter and Vogt, filed Dec. 30, 1998, now U.S. Pat. No. 6,356,903 and incorporated by reference herein.

INT-CL: [07] G06 F 17/00

US-CL-ISSUED: 707/511; 707/203, 707/501.1
US-CL-CURRENT: 707/511; 707/203, 707/501.1

FIELD-OF-SEARCH: 707/10, 707/1, 707/2, 707/3, 707/100, 707/101, 707/104, 707/9, 707/104.1, 707/501.1, 707/203, 707/511, 709/202, 714/100

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

| PAT-NO | ISSUE-DATE | PATENTEE-NAME | US-CL |
|---|---|---|---|
| 5263129 | November 1993 | Ikegaya et al. | |
| 5273434 | December 1993 | Peck | |
| 5311424 | May 1994 | Mukherjee et al. | 705/29 |
| 5317729 | May 1994 | Mukherjee et al. | 707/3 |
| 5455407 | October 1995 | Rosen | 235/380 |
| 5737739 | April 1998 | Shirley et al. | |
| 5842196 | November 1998 | Agarwal et al. | 707/2 |
| 5862346 | January 1999 | Kley et al. | 709/200 |
| 5907837 | May 1999 | Ferrel et al. | 707/3 |
| 5949876 | September 1999 | Ginter et al. | 380/4 |
| 5970476 | October 1999 | Fahey | 705/28 |
| 6067559 | May 2000 | Allard et al. | 709/202 |
| 6078924 | June 2000 | Ainsbury et al. | 707/101 |
| 6085220 | July 2000 | Courts et al. | 709/201 |
| 6119229 | September 2000 | Martinez et al. | 713/200 |
| 6151601 | November 2000 | Papierniak et al. | 707/10 |
| 6157915 | December 2000 | Bhaskaran et al. | 705/7 |
| 6182091 | January 2001 | Pitkow et al. | 707/501 |

OTHER PUBLICATIONS

E. James Whitehead, Jr. "World Wide Web distributed authoring and versioning (WebDAV): an introduction", StandardView vol. 5, No. 1, Mar. 1997, pp. 3-8.*
U.S. patent application Ser. No. 09/222,831, Baxter et al., filed Dec. 30, 1998.
Extensible Markup Language (XML) 1.0, W3C Recommendation Feb. 10, 1998.
Texcel International A.B., Texcel Information Manager Release 2.0 Product Brief, 1998.
Delphi Consulting Group, Delphi Opinion, Astoria 3.0 Chrystal Software, Jun. 1998.
Arbortext, Inc., Introducting Epic.TM. An Enterprise Solution for the Product Information Chain, 1998.
Microstar Software Inc., Content Management Systems.
AIS Software, Application Sheet, Merging XML Content with Relational Databases On-line Catalog Publishing with Dual Prism.
AIS Software, News Release, DUAL PRISM.TM. Now Shipping!, Official Launch of Dual Prism in North America at XML Chicago 1998, Oct. 1998.
AIS Software, Dual Prism.RTM. Powerful Web Content Management and Online Delivery.

ART-UNIT: 2171

PRIMARY-EXAMINER: Metjahic; Safet

ASSISTANT-EXAMINER: Le; Uyen

ABSTRACT:

A system providing views of a technical architecture of an enterprise that take into account the content type interest, level of detail and time frame of desired information. A user, with a browser interface, selects a portal specifying the type of content to be viewed. A profile of the user indicates the level of detail and time frame of information to be obtained. The interface provides the user's selection of desired information within the portal in the form of a page request that is converted into queries of a database that seek content satisfying the type, level of detail and time frame attributes of the request. The content is formed into a web page and provided to the interface by the web server. Content is separated into atomic units allowing the information to be rearranged responsive to each user's needs.

28 Claims, 26 Drawing figures

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 11. Document ID: US 6263332 B1

L9: Entry 11 of 11 File: USPT Jul 17, 2001

US-PAT-NO: 6263332
DOCUMENT-IDENTIFIER: US 6263332 B1

TITLE: System and method for query processing of structured documents

DATE-ISSUED: July 17, 2001

INVENTOR-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY |
|---|---|---|---|---|
| Nasr; Roger I. | Austin | TX | | |
| Webber; Neil | Austin | TX | | |

ASSIGNEE-INFORMATION:

| NAME | CITY | STATE | ZIP CODE | COUNTRY | TYPE CODE |
|---|---|---|---|---|---|
| Vignette Corporation | Austin | TX | | | 02 |

APPL-NO: 09/ 134263
DATE FILED: August 14, 1998

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/5; 707/104, 707/513
US-CL-CURRENT: 707/5; 707/104.1, 707/513

FIELD-OF-SEARCH: 707/1-4.5, 707/513, 707/523

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

| PAT-NO | ISSUE-DATE | PATENTEE-NAME | US-CL |
|---|---|---|---|
| 5138615 | August 1992 | Lamport et al. | 370/400 |
| 5649186 | July 1997 | Ferguson | 707/10 |
| 5655130 | August 1997 | Dodge et al. | 707/511 |
| 5752021 | May 1998 | Nakatsuyama et al. | 707/5 |
| 5778400 | July 1998 | Tateno | 707/513 |
| 5875441 | February 1999 | Nakatsuyama | 707/1 |
| 5893109 | April 1999 | DeRose et al. | 707/104 |
| 5920879 | July 1999 | Kyojima et al. | 707/517 |
| 5940822 | August 1999 | Haderle et al. | 707/3 |
| 5940842 | August 1999 | Sakuta | 707/513 |
| 5956726 | September 1999 | Aoyama et al. | 707/102 |
| 6009436 | December 1999 | Motoyama et al. | 707/102 |
| 6012098 | January 2000 | Bayeh et al. | 709/246 |
| 6018710 | January 2000 | Wynblatt et al. | 704/260 |
| 6023714 | February 2000 | Hill et al. | 707/513 |
| 6047280 | April 2000 | Ashby et al. | 707/2 |
| 6055538 | April 2000 | Kessenich et al. | 707/101 |
| 6072801 | June 2000 | Wood, Jr. et al. | 370/437 |
| 6098071 | August 2000 | Aoyama et al. | 707/102 |
| 6108676 | August 2000 | Nakatsuyama | 707/522 |
| 6128617 | October 2000 | Lowry | 707/100 |

FOREIGN PATENT DOCUMENTS

| FOREIGN-PAT-NO | PUBN-DATE | COUNTRY | US-CL |
|---|---|---|---|
| 0 601 550 A2 | July 1993 | EP | |

OTHER PUBLICATIONS

Marden, Philip, and Munson, Ethan. "Multiple Presentations of WWW Documents Using Style Sheets", Proceedings of the Workshop on New Paradigms in Information Visualization and Manipulation, Nov. 10-14, 1997, pp. 75-78.*
"Microsoft Press Computer Dictionary Third Edition", Definition of the word `tree`, 1997, p. 477.*
Lee, Y.K., Yoo, S-J, and Yoon, K. "Index Structures for Structured Documents", Proceedings of the First ACM International Conference on Digital Libraries, Mar. 20-23, 1996, pp. 91-99.*
Aho, A.V., Hopcraft, J.E., and Ullman, J.D. "Data Structures and Algorithms", Reading, Addison Wesley, 1983. pp. 78-82. QA76.9.D35A38.*
Thompson, Henry S.:"An Introduction to XSL: Slides from a talk to SGML UK" from http://www.itg.ed.ac.uk/.about.ht/swindon.html. Oct. 27, 1997. Sections 5,6,9.*
Lee, Young Kyu: "Querying Structured Documents". System Sciences, 1996., Proceedings of the Twenty-Ninth Annual Hawaii Conference on System Sciences, vol. 2, pp. 155-164.

ART-UNIT: 217

PRIMARY-EXAMINER: Homere; Jean R.

ASSISTANT-EXAMINER: Wassum; Luke S

ABSTRACT:

A computer-implemented method of retrieving information in a first markup language through a query engine and presenting the information in any required markup language. A user inputs a query and may invoke a number of transformative sequences. These sequences contain a markup language pattern and an action, which may include transforming the tags in the first markup language to tags in a different markup language. The appropriate transformative sequence is selected and the pattern from the transformative sequence is compiled. The compiled pattern is used to perform rapid and

efficient searches of documents in the database. A predicate check using the binary coding of the node as well as ancestor information confirms the node. The leaf information associated with a confirmed node is then stored. If necessary, the action from the transformative sequence is applied to change the markup language of the leaf information to that of the user.

36 Claims, 10 Drawing figures



Generate Collection Print

| Term | Documents |
|---|---|
| PARENT.DWPI,TDBD,EPAB,JPAB,USPT,PGPB. | 173200 |
| PARENTS.DWPI,TDBD,EPAB,JPAB,USPT,PGPB. | 11498 |
| CHILD.DWPI,TDBD,EPAB,JPAB,USPT,PGPB. | 116384 |
| CHILDREN.DWPI,TDBD,EPAB,JPAB,USPT,PGPB. | 68795 |
| CHILDRENS.DWPI,TDBD,EPAB,JPAB,USPT,PGPB. | 2859 |
| CHILDS.DWPI,TDBD,EPAB,JPAB,USPT,PGPB. | 4351 |
| RELATIONSHIP.DWPI,TDBD,EPAB,JPAB,USPT,PGPB. | 838370 |
| RELATIONSHIPS.DWPI,TDBD,EPAB,JPAB,USPT,PGPB. | 133990 |
| (7 AND (PARENT NEAR CHILD NEAR RELATIONSHIP)).USPT,PGPB,JPAB,EPAB,DWPI,TDBD. | 11 |
| (L7 AND (PARENT NEAR CHILD NEAR RELATIONSHIP)).USPT,PGPB,JPAB,EPAB,DWPI,TDBD. | 11 |

**Display Format:** - Change Format

Previous Page Next Page

WEST

# Freeform Search

**Database:** US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

**Term:** L7 and (parent near child near relationship)

**Display:** 50 **Documents in Display Format:** - **Starting with Number** 1

**Generate:** ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences | Cases

## Search History

**DATE: Friday, July 26, 2002** Printable Copy Create Case

| Set Name side by side | Query | Hit Count | Set Name result set |
|---|---|---|---|
| *DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR* | | | |
| L9 | L7 and (parent near child near relationship) | 11 | L9 |
| L8 | L7 and (flat near data near structur$) | 1 | L8 |
| L7 | L6 and (data near structur$) | 316 | L7 |
| L6 | ((707/$)!.CCLS.) and xml | 698 | L6 |
| L5 | L1 and xml | 0 | L5 |
| L4 | L1 and (flat near data near structur$) | 0 | L4 |
| L3 | L2 and (flat) | 0 | L3 |
| L2 | L1 and (convert$ and (hierarch$ near data near structur$)) | 11 | L2 |
| L1 | (4954969 5454101 5856925 5924087 6055515 4521882 4928247 4995112 5224100 5303367 5392220 5430870 5438511 5842213 5923881 5960433 6112209 4555771 4796199 4864497 4947447 5251322 5265244 5280547 5295261 5341466 5363320 5394249 5410675 5467471 5490258 5495440 5506788 5542085 5561608 5577239 5627999 5634123 5636158 5648813 5651101 5664177 5668965 5701467 5703800 5761637 5771384 5784526 5799183 5808901).pn. | 135 | L1 |

END OF SEARCH HISTORY

WEST

# Freeform Search

**Database:** US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

**Term:** L7 and (parent near child near relationship)

**Display:** 50 **Documents in Display Format:** - **Starting with Number** 1

**Generate:** ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences | Cases

## Search History

**DATE: Friday, July 26, 2002** Printable Copy Create Case

| Set Name side by side | Query | Hit Count | Set Name result set |
|---|---|---|---|
| *DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR* | | | |
| L9 | L7 and (parent near child near relationship) | 11 | L9 |
| L8 | L7 and (flat near data near structur$) | 1 | L8 |
| L7 | L6 and (data near structur$) | 316 | L7 |
| L6 | ((707/$)!.CCLS.) and xml | 698 | L6 |
| L5 | L1 and xml | 0 | L5 |
| L4 | L1 and (flat near data near structur$) | 0 | L4 |
| L3 | L2 and (flat) | 0 | L3 |
| L2 | L1 and (convert$ and (hierarch$ near data near structur$)) | 11 | L2 |
| L1 | (4954969 5454101 5856925 5924087 6055515 4521882 4928247 4995112 5224100 5303367 5392220 5430870 5438511 5842213 5923881 5960433 6112209 4555771 4796199 4864497 4947447 5251322 5265244 5280547 5295261 5341466 5363320 5394249 5410675 5467471 5490258 5495440 5506788 5542085 5561608 5577239 5627999 5634123 5636158 5648813 5651101 5664177 5668965 5701467 5703800 5761637 5771384 5784526 5799183 5808901).pn. | 135 | L1 |

END OF SEARCH HISTORY

WEST

**End of Result Set**

Generate Collection Print

L2: Entry 11 of 11 File: DWPI Mar 15, 1994

DERWENT-ACC-NO: 1994-091836
DERWENT-WEEK: 199411


TITLE: Data record to hybrid structure conversion method for database - storing navigational data in topological tree structure with each leaf node specifying unique record in relational informational database

Patent Serial Number (1):
5295261

Basic Abstract Text (1):
The method involves separating the number of fields into navigational fields and informational fields, where geographic values encompassed in each navigational field of the data records are of a geographical class defined by the field type of the navigational field. The navigationa l fields are ordered into a number of distinct hierarchical data structures each comprising a number of node classes and links. Among the number of distinct hierarchical data structures one of the distinct hierarchical data structures is identified having at least one geographica l class in common. Portions of the distinct hierarchical data structures having one or more geographic classes in common are merged by linking the structures to a single node class.

Basic Abstract Text (2):
For each the node class in the merged hierarchical data structure a table is created comprising an identifier describing the geographical class represented by the node class, an enumeration list encompassing the geographical values of the node class, and a number of linkage structures.

Standard Title Terms (1):
DATA RECORD HYBRID STRUCTURE CONVERT METHOD DATABASE STORAGE NAVIGATION DATA TOPOLOGICAL TREE STRUCTURE LEAF NODE SPECIFIED UNIQUE RECORD RELATED DATABASE

WEST

Generate Collection | Print

**Search Results** - Record(s) 1 through 11 of 11 returned.

☐ 1. Document ID: US 20020091702 A1

L9: Entry 1 of 11 File: PGPB Jul 11, 2002

PGPUB-DOCUMENT-NUMBER: 20020091702
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020091702 A1

TITLE: Dynamic object-driven database manipulation and mapping system

PUBLICATION-DATE: July 11, 2002

INVENTOR-INFORMATION:

| NAME | CITY | STATE | COUNTRY | RULE-47 |
|---|---|---|---|---|
| Mullins, Ward | San Francisco | CA | US | |

US-CL-CURRENT: 707/100

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 2. Document ID: US 20020083034 A1

L9: Entry 2 of 11 File: PGPB Jun 27, 2002

PGPUB-DOCUMENT-NUMBER: 20020083034
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020083034 A1

TITLE: Method and apparatus for extracting data objects and locating them in virtual space

PUBLICATION-DATE: June 27, 2002

INVENTOR-INFORMATION:

| NAME | CITY | STATE | COUNTRY | RULE-47 |
|---|---|---|---|---|
| Orbanes, Julian | Cambridge | MA | US | |
| Guzman, Adriana | Cambridge | MA | US | |

US-CL-CURRENT: 707/1

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

☐ 3. Document ID: US 20020078054 A1

L9: Entry 3 of 11 File: PGPB Jun 20, 2002

PGPUB-DOCUMENT-NUMBER: 20020078054
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020078054 A1

TITLE: Group forming system, group forming apparatus, group forming method, program, and medium

PUBLICATION-DATE: June 20, 2002

INVENTOR-INFORMATION:

| NAME | CITY | STATE | COUNTRY | RULE-47 |
|---|---|---|---|---|
| Kudo, Takahiro | Osaka | | JP | |
| Matsuura, Satoshi | Kyotanabe-shi | | JP | |
| Ozawa, Jun | Nara-shi | | JP | |

US-CL-CURRENT: 707/10

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 4. Document ID: US 20020069215 A1

L9: Entry 4 of 11 File: PGPB Jun 6, 2002

PGPUB-DOCUMENT-NUMBER: 20020069215
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020069215 A1

TITLE: Apparatus for viewing information in virtual space using multiple templates

PUBLICATION-DATE: June 6, 2002

INVENTOR-INFORMATION:

| NAME | CITY | STATE | COUNTRY | RULE-47 |
|---|---|---|---|---|
| Orbanes, Julian | Cambridge | MA | US | |
| Guzman, Adriana | Cambridge | MA | US | |

US-CL-CURRENT: 707/500

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 5. Document ID: US 20020035555 A1

L9: Entry 5 of 11 File: PGPB Mar 21, 2002

PGPUB-DOCUMENT-NUMBER: 20020035555
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020035555 A1

TITLE: System and method for building and maintaining a database

PUBLICATION-DATE: March 21, 2002

INVENTOR-INFORMATION:

| NAME | CITY | STATE | COUNTRY | RULE-47 |
|---|---|---|---|---|
| Wheeler, David B. | Austin | TX | US | |
| Wotring, Steven C. | Austin | TX | US | |

US-CL-CURRENT: 707/1; 707/104.1

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 6. Document ID: US 20010049688 A1

L9: Entry 6 of 11 File: PGPB Dec 6, 2001

PGPUB-DOCUMENT-NUMBER: 20010049688
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010049688 A1

TITLE: System and method for providing an intelligent multi-step dialog with a user

PUBLICATION-DATE: December 6, 2001

INVENTOR-INFORMATION:

| NAME | CITY | STATE | COUNTRY | RULE-47 |
|---|---|---|---|---|
| Fratkina, Raya | Yorktown Heights | NY | US | |
| Anderson, Monica | San Jose | CA | US | |
| Angel, Mark A. | Napa | CA | US | |
| Copperman, Max | Santa Cruz | CA | US | |
| Huffman, Scott B. | Redwood City | CA | US | |
| Kay, David | Los Gatos | CA | US | |
| Stern, Robert | Cupertino | CA | US | |

US-CL-CURRENT: 707/104.1; 345/968

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 7. Document ID: US 20010037345 A1

L9: Entry 7 of 11 File: PGPB Nov 1, 2001

PGPUB-DOCUMENT-NUMBER: 20010037345
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010037345 A1

TITLE: Tagging XML query results over relational DBMSs

PUBLICATION-DATE: November 1, 2001

INVENTOR-INFORMATION:

| NAME | CITY | STATE | COUNTRY | RULE-47 |
|---|---|---|---|---|
| Kiernan, Gerald George | San Jose | CA | US | |
| Shanmugasundaram, Jayavel | San Jose | CA | US | |
| Shekita, Eugene Jon | San Jose | CA | US | |

US-CL-CURRENT: 707/513

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 8. Document ID: US 20010018697 A1

L9: Entry 8 of 11 File: PGPB Aug 30, 2001

PGPUB-DOCUMENT-NUMBER: 20010018697
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010018697 A1

TITLE: Structured document processing system and structured document processing method

PUBLICATION-DATE: August 30, 2001

WEST

Generate Collection Print

L2: Entry 10 of 11 File: USPT

DOCUMENT-IDENTIFIER: US 4555771 A
TITLE: Data processing system for data base management of matrix type data

US PATENT NO. (1):
4555771

Brief Summary Text (9):
Another object of the invention is to provide a data processing system for virtually converting matrix type data to a desired hierarchical structure data.

Brief Summary Text (13):
wherein a1, a2 . . . an is the n-dimensional coordinate and A1, A2 . . . An is the coordinate axis size. The calculator includes a converter connected to the scheme code storage, the address storages and the designating device for providing correspondence between the contents of the designated one of the address storages and the n-dimensional coordinate in accordance with the contents of the scheme code storage, the converter having an output, and a linear expander connected to the n-coordinate axis size storage, the designating device and the converter for executing operation of the one-dimensional address information value in accordance with the output of the converter and the contents of the n-coordinate axis size storage.

Drawing Description Text (14):
FIG. 8F is a block diagram of an embodiment of a converter unit CV;

Detailed Description Text (43):
FIG. 8F is a block diagram of an embodiment of a converter unit CV, which distributes signal inputs supplied to input terminals I1, I2 and I3 to output terminals O1, O2 and O3 in accordance with an input supplied to terminal C when a pulse is provided at terminal E. More particularly, the input and output terminals of the converter unit are connected, for example, I1 to O1, I2 to O2 and I3 to O3 when a specific value is applied to the terminal C, and are connected, for example, I1 to O2, I2 to O3 and and I3 to O1 when a different value is applied to the terminal C. When a terminal D is driven, an inverted replacement is provided. The converter unit CV is operated with a delay of one cycle from the driving pulse. When "O" is applied to any of the input terminals I1, I2 and I3, a terminal N is driven. I1 to I3, O1 to O3 and C are multi-bit signals and E and N are single-bit signals.

Detailed Description Text (44):
The converter unit CV of FIG. 8F includes a decoder 601, AND gates 602 to 610, OR gates 611 to 615, latches 616 to 618, a delay circuit 619 for a one cycle delay and zero detect circuits 620 to 622. When the input C equals zero, for example, the decoder 601 drives its output line L1, and accordingly the input terminals I1, I2 and I3 are connected to the OR gates 611, 612 and 613 via the AND gates 602, 605 and 608, respectively.

Detailed Description Text (52):
In FIG. 9, 101 to 119 are register units R, as shown in FIG. 8C, 120 to 122 are arithmetic units AU as shown in FIG. 8A, 123 to 128 are bypass gates BG, as shown in FIG. 8B, 129 to 131 are converter units CV, as shown in FIG. 8F, 132 is a set selection unit SS, as shown in FIG. 8D, 133 and 134 are OR gates OR, as shown in FIG. 8E, 136 and 137 are NOT circuits NOT, as shown in FIG. 8H, 135 and 138 are AND gates AND, as shown in FIG. 8I, 139 is a linear expander LE, as shown in FIG. 8G and 140 is a compose matrix unit CM, as shown in FIG. 8J.

Detailed Description Text (53):
The registers 101, 102 and 103 store the sizes A1, A2 and A3, respectively, of the

three coordinates a1, a2 and a3, respectively. In FIG. 1, specifically, A1=2, A2=2, A3=3. The register 119 stores the scheme code C, which designates the desired hierarchical structure. The converter 129 replaces the size information A1 to A3 received from the registers 101 to 103, respectively, and distributes such information to the registers 104 to 106, respectively, in accordance with the code C provided by the register 119. When the scheme in FIG. 4 is designated, for example, A2 is stored in the register 104, A3 is stored in the register 105 and A1 is stored in the register 106.

Detailed Description Text (58):
The code for designating the third set SS3 is then fed to the set selection unit 132 as the code S in order to operate the arithmetic unit 122. The contents of the register 109 are thus incremented by the modulus on the basis of the contents of the register 106. When the process is completed, a pulse is provided at the terminal K of the arithmetic unit 122, and is fed to the terminal E of the converter 130 and to the terminal Q of the linear expander 139. The converter 130 transfers the signals I1, I2 and I3 received from the registers 109, 111 and 112 to its outputs 01, 02 and 03, in accordance with the scheme designation code C received from the register 119. The linear expander 139 provides the aforedescribed process. In the current state, access is to the #3 record (113) in FIG. 4, for example. In other words, I1=1, I2=1 and I3=3 as the inputs of the linear expander 139, and A1=2, A2=2 and A3=3. Therefore, ##EQU2## More particularly, the one-dimensional address #3 may be obtained from the three-dimensional coordinate (113).

Detailed Description Text (64):
It is possible to designate a specific member record of the other owner record in a known data base management system under the condition that a specific member record is designated under a specific owner record. The location information of the new owner record 0' may be set in the register 117. The one-dimensional address P' of a new member record may be set in the register 118. The contents of the register 118 are converted to a three-dimensional coordinate by the compose matrix unit 140. Each coordinate value of such three-dimensional coordinate is replaced by the converter 131 in accordance with the scheme designation code in the register 119 and set in each register 109, 111 and 112.

Detailed Description Text (66):
Furthermore, when the converter unit 130 detects that any of the inputs I1 to I3 is zero, a pulse is provided at its terminal N and fed to the terminal I2 of the AND gate 135. When the contents of any of the registers 109, 111 and 112 becomes zero, therefore, under the condition that a pulse is not output at the terminal E of the arithmetic unit 122, a signal N may be output at the terminal O of the AND gate 135. The signal N is an error signal.

CLAIMS:

calculating means including a converter means including a converter means, said converter means connected to said n-coordinate axes, said scheme code storage means, said address storage means and said selecting means for calculating the one-dimensional address from the contents of said address storage means, the contents of said scheme code storage means and the contents of said n-coordinate axes, said file means being connected to said calculating means and being accessed by the one-dimensional address output from said calculating means, said converter means providing correspondence between the contents of said selected one of said address storage means and said n-dimensional coordinate in accordance with the contents of said scheme code storage means said converter means including a decoder, responsive to input scheme code signals, for providing different outputs of said converter means for the same inputs from said n-coordinate-axes, said converter means providing an output, and linear expander means for executing operation of said one-dimensional address information value in accordance with the output of said converter means and the contents of said n-coordinate axes, said linear expander means connected to said n-coordinate axes, said selecting means, and said converter means.

WEST

Generate Collection Print

L2: Entry 6 of 11 File: USPT

DOCUMENT-IDENTIFIER: US 5438511 A
TITLE: Disjunctive unification

US PATENT NO. (1):
5438511

Brief Summary Text (3):
A number of general approaches have been proposed for solving a system such as a logical formula whose terms are equations. Some of these, such as term rewriting, operate directly on the terms, which are typically conjoined equations. Others, such as unification or merging, operate on terms converted into suitable data structures. Approaches of either of these types typically perform poorly if the system being solved includes disjunctions.

Brief Summary Text (18):
A closely related aspect of the invention is based on the recognition of a problem in associating context identifiers within a hierarchical data structure. In general, the unification of different context identifiers results in an expansion or cross product multiplication to consider all possible combinations of contexts, similar to expanding to the disjunctive normal form. Therefore, context identifiers that are relatively high in hierarchical data structures being unified result in an inefficient expansion between the context identifiers in the data structures.

Brief Summary Text (19):
This aspect of the invention solves this problem based on the recognition that the number of context identifiers that are relatively high in a hierarchical data structure can be reduced by combining context identifiers. Context identifiers can only be combined, however, if data units that depend from them can be combined. Therefore, this solution depends on including context identifiers in data structures such that the data units that depend from them can frequently be combined, either by unification or other techniques.

Detailed Description Text (17):
FIG. 1 shows a sequence of general steps, beginning with disjunctive system 10 that includes disjunctions 12 and 14, each illustratively including a number of feature-value equations although the invention is also applicable to other types of disjunctions, as described above. Each disjunction is converted to a data structure, and these data structures are then unified to obtain a solution of the disjunctive system.

Detailed Description Text (21):
The conversion of each disjunction into a data structure for unification could be done in various ways. For example, data structures 22 and 24 are hierarchical data structures that include substantially all the information in disjunctions 12 and 14, respectively. In other words, it is possible in general to map disjunction 12 onto data structure 22 and disjunction 14 onto data structure 24, and vice versa. Since all the equations in each disjunction have the same function, represented by the function variable token "f", the uppermost hierarchical level that is required in each data structure is that of the function; a higher root level could be provided if there were more than one function token. The next lower hierarchical level in each data structure includes a context identifier identifying each disjunct, in accordance with the invention. The disjuncts of disjunction 12 are identified as "c.sub.1 " and "c.sub.2 ", while those of disjunction 14 are identified as "c.sub.3 " and "c.sub.4 ". These context identifiers correspond to genuine alternative contexts. As discussed above and as illustrated below, these context identifiers make it possible to preserve information about combinations of disjuncts within data structures being unified.

Finally, the lowest level of each data structure includes feature-value pairs, each feature-value pair including a feature token and a value token. The feature-value pairs within each disjunct depend from that disjunct's context identifier.

Detailed Description Text (22):
As discussed above, unifying data structures 22 and 24 would be inefficient, because the context identifiers are relatively high in the hierarchy of each data structure, which would result in expansion to consider all possible combinations of context identifiers. Therefore, in accordance with a further aspect of the invention, alternative context-value pairs in data structures 22 and 24 are combined to produce data structures 32 and 34, in which each feature that previously occurred in a feature-value pair now has a disjunctive value data unit that includes appropriate context identifiers, so that the feature tokens are now above the context identifiers in the hierarchy.

Detailed Description Text (40):
SOLVEAND handles each conjunct according to its relation, which may illustratively be equality or disjunction. The step in box 102 branches based on this relation. If the relationship is equality, SOLVEAND calls LOCATE to find the current values of the arguments and to convert equations into data structures; SOLVEAND then calls UNIFY to unify the data structures corresponding to the current values of the arguments, in box 104. In converting to data structures, LOCATE may in turn call LOCATE.VAR or LOCATE.AE, either of which may call CREATECONTEXTVALUE to create an entity of type Disjunction, which is a set of context-value pairs and therefore is a disjunctive value data unit as described above. If, rather than equality, the relationship is disjunction, SOLVEAND calls SOLVEOR to handle it, which begins an inner iterative loop in box 110. When unification in box 104 is complete or all the disjuncts have been handled in the inner iterative loop, SOLVEAND begins another iteration with the next conjunct, in box 100.

Detailed Description Text (52):
FIG. 6B shows f-structure 300 which is equivalent to disjunctive value 200, but within which each feature has a disjunctive value. The f-structure 300 may be thought of as a reduced form of disjunctive value 200. It could be obtained by converting disjunctive value 200 to its reduced form by combining the context-value pairs shown in FIG. 6A, a technique discussed in greater detail below; it could alternatively be obtained directly from disjunction (1). Like disjunctive value 200, it has outer brackets 302 and 304, but rather than having three terms within its outer brackets, f-structure 300 has two terms-one each for the two feature tokens A and B that occur in disjunction (1). In addition to a respective feature token, each of the terms within outer brackets 302 and 304 also includes a set of inner brackets 310 and 312. The inner bracket in each term contains a disjunctive value for the respective feature token.

Detailed Description Text (78):
The test in box 410 determines whether D1 and D2 are of type Disjunction. If either entity is not of type Disjunction, the step in box 412 calls COERCETODISJUNCTION to convert it into an entity of type Disjunction, which can be done by copying the non-disjunctive entity and then smashing the original to be an entity of type Disjunction with a single context-value pair, the context being the NULL context and the value being the copy of the non-disjunctive entity.

Detailed Description Text (83):
When all the pairs in D2 have been handled, UNIFY.CONTEXTS returns to the test in box 420 to handle the next context-value pair in D1. When all the pairs in D1 have been handled, the step in box 450 calls REDUCEDISJUNCTION, discussed in more detail below, to convert the unified version of D1 and D2 to the reduced form with disjunctive values. Then, the test in box 452 determines whether a swap was performed in box 416. If not, D1 is returned, in box 406, but if so, D2 is returned in box 454.

Detailed Description Text (135):
Although the invention does not solve the NP-complete problem, it does avoid exponential growth for typical examples of the problem that are decomposible into independent P class problems. For example, in a long sentence, different disjunctions are typically independent if far apart, so that use of the disjunctive normal form is unnecessary and the disjunctions can be treated independently as P class problems. In cases where disjunctions are not independent, the use of context identifiers keeps track of their relationships and allows the consideration of combinations of disjuncts from different disjunctions as necessary without solving the NP-complete problem. Since any NP-complete problem can be converted into any other, the invention should be useful with other problems in which disjunctions occur that can be treated as independent P

class problems.

WEST

Generate Collection Print

L2: Entry 9 of 11 File: USPT

DOCUMENT-IDENTIFIER: US 5295261 A
TITLE: Hybrid database structure linking navigational fields having a hierarchial database structure to informational fields having a relational database structure

US PATENT NO. (1):
5295261

Drawing Description Text (3):
FIGS. 2(A), 2(B), and 2(C) illustrate the manner in which one of the tables shown in FIG. 1 can be converted into a database according to the present invention.

Detailed Description Text (17):
The final step of converting the database consists of replacing table 41 by a tree structure 50 as shown in FIG. 2(C). The nodes in tree structure 50 are divided into sets shown at 51-54. Each set of nodes corresponds to a column in table 41. The nodes in set 51 correspond to the state, those in set 52 correspond to the city, those in set 53 correspond to the street address, and those in set 54 correspond to the unique identifier defined for each record in the original database. Each node in a given set is linked to a node in a set one level higher up in the hierarchy. Hence, there is no need to repeat the redundant data. For example, by traversing the tree from any given city node to the state node to which it is linked, one may ascertain the state in which the city is located.

CLAIMS:

1. A method for converting organized data from a data record structure into a hybrid structure which allows for efficient storage and retrieval of the organized data by a digital computer, where the data record structure includes a plurality of records each divided into a plurality of fields, where each field of the plurality of fields stores data of a field type associated with the field and where the organized data in the data record structure is encompassed in the plurality of fields of the plurality of records, the method comprising the steps of:

ordering said navigational fields into a plurality of distinct hierarchical data structures each comprising a plurality of node classes and links therebetween, where each said node class corresponds to a specific geographical class and each node in said node class corresponds to a distinct geographic value of said geographical class, where a node class lower in said hierarchical data structure represents a geographical class that is hierarchically subordinate to and wholly encompassed in the geographical classes represented by node classes higher in said hierarchical data structure, and where said links between said node classes indicate which geographic values listed in a lower node class are geographically encompassed in a geographic value listed in a higher node class immediately adjacent to and above said lower node class in said hierarchical data structure;

identifying among the plurality of distinct hierarchical data structures one of said distinct hierarchical data structures having at least one geographical class in common, where a distinct hierarchical data structure is one which is hierarchically independent from all other hierarchical data structures and which has at least one geographic class that is unique to that hierarchical data structure;

merging portions of said distinct hierarchical data structures having one or more geographic classes in common by linking said structures to a single node class for each said geographical class said structures have in command; and

creating for each said node class in the merged hierarchical data structure a table

comprising an identifier describing the geographical class represented by the node class, an enumeration list encompassing the geographical values of said node class, and a plurality of linkage structures, where each linkage structure comprises an adjacent class identifier describing a geographical class immediately adjacent to and above said node class in said merged hierarchical data structure, and an adjacent link list, specifying for each first geographic value in said enumeration list, a second geographic value in an adjacent node class in which said first geographic value is geographically encompassed, and wherein the navigational field of a lowest geographical class includes said unique identifier thereby pointing to a corresponding one of said informational fields containing said same unique identifier.

2. The method of claim 1, wherein said step of ordering the navigational fields into a plurality of hierarchical data structures further comprises the steps of:

creating a lowest node class for each hierarchical data structure, where said lowest node class encompasses nodes representing the unique identifiers; and

linking said lowest node class to the node class lowest in the hierarchical data structure which encompasses geographic values to indicate a relationship between said geographic values and the records of said informational fields.

3. A method for converting organized data from a data record structure into a hybrid structure which allows for efficient storage and retrieval of the organized data by a digital computer, where the data record structure includes a plurality of records each divided into a plurality of fields, where each field of the plurality of fields stores data of a field type associated with the field and where the organized data in the data record structure is encompassed in the plurality of fields of the plurality of records, the method comprising the steps of:

ordering said navigational fields into a plurality of distinct hierarchical data structures each comprising a plurality of node classes and links therebetween, where each said node class corresponds to a specific class of values and each node in said node class corresponds to a distinct value of said class, where a node class lower in said hierarchical data structure represents a class that is hierarchically subordinate to and wholly encompassed in the classes represented by node classes higher in said hierarchical data structure, and where said links between said node classes indicate which values listed in a lower node class are specifically related to a value listed in a higher node class immediately adjacent to and above said lower node class in said hierarchical data structure;

identifying among the plurality of distinct hierarchical data structures ones of said distinct hierarchical data structures having at least one class in common, where a distinct hierarchical data structure is one which is hierarchically independent from all other hierarchical data structures and which has at least one class that is unique to that hierarchical data structure;

merging portions of said distinct hierarchical data structures having one or more classes in common by linking said structures to a single node class for each class said structures have in common, where each said node in the merged node class represents the unique occurrence of a value encompassed in the common, yet unmerged node classes; and

creating for each said node class in the merged hierarchical data structure a table comprising an identifier describing the class represented by the node class, an enumeration list encompassing the values of said node class, and a plurality of linkage structures, where each linkage structure comprises an adjacent class identifier describing a class immediately adjacent to and above said node class in said merged hierarchical data structure, and an adjacent link list, specifying for each first value in said enumeration link, a second value in an adjacent node class to which said first value is related, and wherein the navigational field of a lowest class includes said unique identifier thereby pointing to a corresponding one of said informational fields containing said same unique identifier.

4. The method of claim 3, wherein said step of ordering the navigational fields into a plurality of hierarchical data structures further comprises the steps of:

creating a lowest node class for each hierarchical data structure, where said lowest node class encompasses nodes representing the unique identifiers; and

linking said lowest node class to the node class lowest in the hierarchical data

structure which encompasses a class of specific values to indicate a relationship between said specific values and the records of said informational fields.